

PATENT
8001-1178

IN THE U.S. PATENT AND TRADEMARK OFFICE

In re application of: Takao AIGO
Appl. No.: NEW NON-PROVISIONAL
Filed: January 5, 2004
Title: DISK ARRAY CONTROL APPARATUS AND METHOD

Conf.:

Group:

Examiner:

CLAIM TO PRIORITY

Assistant Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

January 5, 2004

Sir:

Applicant(s) herewith claim(s) the benefit of the priority filing date of the following application(s) for the above-entitled U.S. application under the provisions of 35 U.S.C. § 119 and 37 C.F.R. § 1.55:

<u>Country</u>	<u>Application No.</u>	<u>Filed</u>
JAPAN	2003-039185	February 18, 2003

Certified copy(ies) of the above-noted application(s) is(are) attached hereto.

Respectfully submitted,

YOUNG & THOMPSON



Eric Jensen, Reg. No. 37,855

745 South 23rd Street
Arlington, VA 22202
Telephone (703) 521-2297

EJ/ia

Attachment(s): 1 Certified Copy(ies)

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 2 月 1 8 日
Date of Application:

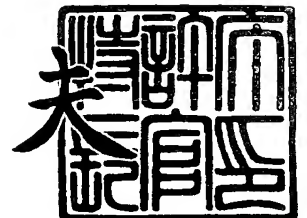
出 願 番 号 特 願 2 0 0 3 - 0 3 9 1 8 5
Application Number:
[ST. 10/C] : [J P 2 0 0 3 - 0 3 9 1 8 5]

出 願 人 日 本 電 気 株 式 有 限 公 司
Applicant(s):

2 0 0 3 年 1 0 月 2 日

特 許 庁 長 官
Commissioner,
Japan Patent Office

今 井 康 夫



出 証 番 号 出 証 特 2 0 0 3 - 3 0 8 1 3 1 7

【書類名】 特許願

【整理番号】 67000115

【提出日】 平成15年 2月18日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F

【発明の名称】 ディスクアレイ制御装置及び方法並びにディスクアレイ
制御プログラム

【請求項の数】 9

【発明者】

 【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

 【氏名】 相合 孝雄

【特許出願人】

 【識別番号】 000004237

 【氏名又は名称】 日本電気株式会社

【代理人】

 【識別番号】 100079164

 【弁理士】

 【氏名又は名称】 高橋 勇

 【電話番号】 03-3862-6520

【手数料の表示】

 【予納台帳番号】 013505

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9003064

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ディスクアレイ制御装置及び方法並びにディスクアレイ制御プログラム

【特許請求の範囲】

【請求項 1】 ホストコンピュータから出力された I/O 処理要求に基づき、キャッシュメモリを有するディスクアレイに対してマルチタスク処理を実行する手段と、各タスク処理の実行に必要なプログラムや制御情報を使用頻度の高いものに入れ換えつつ保持する記憶部とを備えたディスクアレイ制御装置において、

前記ホストコンピュータからの I/O 処理要求を受信するホスト I/O 受信手段と、

低優先タスク処理を実行する低優先度 I/O 処理実行手段と、

前記低優先タスク処理よりも優先的に高優先タスク処理を実行する高優先度 I/O 処理実行手段と、

前記ホスト I/O 受信手段で受信された I/O 処理を、実行中の前記高優先タスク処理の数が一定数未満の場合は高優先タスク処理とし、実行中の前記高優先タスク処理の数が前記一定数に一致する場合は低優先タスク処理とする実行タスク選択手段と、

を備えたことを特徴とするディスクアレイ制御装置。

【請求項 2】 前記低優先タスク処理の実行直後に当該低優先タスク処理を前記高優先タスク処理に変更するタスク優先度変更手段、

を更に備えた請求項 1 記載のディスクアレイ制御装置。

【請求項 3】 実行されたタスク処理が前記キャッシュメモリに対するものである場合にキャッシュヒットと判定するキャッシュヒット判定手段と、

一定時間内に実行されたタスク処理の総数に占める前記キャッシュヒットの数にキャッシュヒット率とし、このキャッシュヒット率が高ければ実行するタスク処理数を少なくし当該キャッシュヒット率が低ければ実行するタスク処理数を多くするキャッシュヒット率監視手段と、

を更に備えた請求項 1 又は 2 記載のディスクアレイ制御装置。

【請求項 4】 ホストコンピュータから出力された I/O 処理要求に基づき、キャッシュメモリを有するディスクアレイに対してマルチタスク処理を実行する手段と、各タスク処理の実行に必要なプログラムや制御情報を使用頻度の高いものに入れ換えつつ保持する記憶部とを備えたディスクアレイ制御装置に用いられ、

前記ホストコンピュータから I/O 処理要求を受信し、

この I/O 処理を、実行中の高優先タスク処理の数が一定数未満の場合は高優先タスク処理とし、実行中の前記高優先タスク処理の数が前記一定数に一致する場合は低優先タスク処理とし、

この低優先タスク処理よりも優先的に前記高優先タスク処理を実行する、
ディスクアレイ制御方法。

【請求項 5】 前記低優先タスク処理の実行直後に当該低優先タスク処理を前記高優先タスク処理に変更する、

請求項 4 記載のディスクアレイ制御方法。

【請求項 6】 実行されたタスク処理が前記キャッシュメモリに対するものである場合にキャッシュヒットと判定し、

一定時間内に実行されたタスク処理の総数に占める前記キャッシュヒットの数をキャッシュヒット率とし、このキャッシュヒット率が高ければ実行するタスク処理数を少なくし当該キャッシュヒット率が低ければ実行するタスク処理数を多くする、

請求項 4 又は 5 記載のディスクアレイ制御方法。

【請求項 7】 ホストコンピュータから出力された I/O 処理要求に基づき、キャッシュメモリを有するディスクアレイに対してマルチタスク処理を実行するマイクロプロセッサと、各タスク処理の実行に必要なプログラムや制御情報を使用頻度の高いものに入れ換えつつ保持する記憶部とを備えたディスクアレイ制御装置に用いられ、

前記ホストコンピュータからの I/O 処理要求を受信するホスト I/O 受信手段、

低優先タスク処理を実行する低優先度 I/O 処理実行手段、

前記低優先タスク処理よりも優先的に高優先タスク処理を実行する高優先度 I/O 処理実行手段、及び、

前記ホスト I/O 受信手段で受信された I/O 処理を、実行中の前記高優先タスク処理の数が一定数未満の場合は高優先タスク処理とし、実行中の前記高優先タスク処理の数が前記一定数に一致する場合は低優先タスク処理とする実行タスク選択手段、

として前記マイクロプロセッサを機能させるディスクアレイ制御プログラム。

【請求項 8】 前記低優先タスク処理の実行直後に当該低優先タスク処理を前記高優先タスク処理に変更するタスク優先度変更手段、

として前記マイクロプロセッサを更に機能させる請求項 7 記載のディスクアレイ制御プログラム。

【請求項 9】 実行されたタスク処理が前記キャッシュメモリに対するものである場合にキャッシュヒットと判定するキャッシュヒット判定手段、及び

一定時間内に実行されたタスク処理の総数に占める前記キャッシュヒットの数をキャッシュヒット率とし、このキャッシュヒット率が高ければ実行するタスク処理数を少なくし当該キャッシュヒット率が低ければ実行するタスク処理数を多くするキャッシュヒット率監視手段、

として前記マイクロプロセッサを更に機能させる請求項 7 又は 8 記載のディスクアレイ制御プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ディスク装置を複数台並列に接続し、全体をまとめて制御することにより、データ転送速度の高速化及びデータ障害の耐久性の向上を図るディスクアレイに関し、詳しくはディスクアレイに用いられる制御装置等に関する。

【0002】

【従来の技術】

従来、ディスクアレイでは、ホストコンピュータからの書き込み／読み出し要求に対する処理能力を向上させるため、多数の書き込み／読み出し処理（以下「

I/O処理」という。)を同時に実行している。これを実現するために、ディスクアレイは、ファームウェアにリアルタイムOS等を使用することにより、マルチタスク処理を行う場合がある(例えば、下記特許文献1, 2参照。)。このマルチタスク処理によれば、それぞれのタスクが1つのI/O処理を実行し、また複数のタスクが同時に処理を実行できるため、単位時間あたりのI/O処理能力が向上する。一般にリアルタイムOSのマルチタスク処理では、タスク毎にそれぞれ制御用のメモリ領域をローカルメモリに持つことにより、複数のタスクを同時に実行できるようになっている。

【0003】

一方、近年のディスクアレイでは、マイクロプロセッサにプロセッサキャッシュを搭載することにより、性能の向上を図っている。つまり、ローカルメモリの同じメモリ領域を何度もアクセスする場合に、アクセス速度の速いプロセッサキャッシュ上にデータ(プログラムや制御情報など)を保持しておき、2回目のアクセス以降ではプロセッサキャッシュ上のデータを使用する。これにより、ローカルメモリへのアクセス回数が減るので、処理速度が向上する。プロセッサキャッシュへのデータ格納方式としては、一般にLRU(Least Recently Used)方式が使用されている。このLRU方式とは、プロセッサキャッシュにデータを格納するブロックがない場合に、プロセッサキャッシュ上で最後に参照されてから現在までの経過時間が最も長いデータブロックを追い出す方式である。

【0004】

なお、「キャッシュメモリ」とは、ホストコンピュータからディスク装置に対して書き込み/読み出しされたデータの中でよく使うデータを保持しておくものである。これに対して、「プロセッサキャッシュ」とは、プロセッサが動作するためのマイクロプログラムや制御情報を格納しておくプロセッサ用のメモリ(ローカルメモリ)に格納されたデータの中でよく使うデータを保持しておくものである。また、ここでいう「I/O処理」とは、書き込み処理又は読み出し処理のことである。

【0005】

【特許文献1】

特開平 2-287849 号公報

【特許文献 2】

特開平 5-298122 号公報

【0006】

【発明が解決しようとする課題】

マルチタスク処理では、タスク毎にメモリ領域を割り当てている（タスクメモリ）。そのため、タスク数が少ない場合には、全てのタスクメモリをプロセッサキャッシュ内に格納できることにより、プロセッサキャッシュのみにアクセスすればよいので、高速に処理することができる。しかし、タスク数が増えると、全てのタスクメモリをプロセッサキャッシュ内に格納できなくなるため、プロセッサキャッシュのデータに対して追い出しや再読み込みが発生することにより、処理速度が低下してしまう。一方、ディスクアレイにおいては、ホストコンピュータからの I/O 処理要求を多重で実行する必要があるため、多数のタスクによって処理しなければ、十分な処理速度を得ることができない。

【0007】

【発明の目的】

そこで、本発明の目的は、ディスクアレイの処理速度を向上できるディスクアレイ制御装置等を提供することにある。

【0008】

【課題を解決するための手段】

ホストコンピュータからの読み出し要求時にディスクアレイのキャッシュメモリ内にデータが存在する場合（キャッシュヒット時）は、ディスク装置にアクセスする必要がないので、高速に処理を実行できる。そのため、タスク数は少ない方がプロセッサキャッシュを有効に利用できる。なぜなら、タスク数が少なければ全てのタスクメモリをプロセッサキャッシュ内に格納できるため、プロセッサキャッシュのみにアクセスすればよいので、高速に処理できるからである。

【0009】

一方、読み出し要求時にディスクアレイのキャッシュメモリ内にデータが存在しないためディスク装置からデータを読み出すような場合（キャッシュミス時）

は、一つの処理に時間がかかるため、タスク数が十分でない全体としての処理速度が低下してしまう。つまり、プロセッサキャッシュが有効に利用されていない時にタスク数を減らしても、プロセッサキャッシュによる処理速度の向上が望めないばかりか、却って処理速度が低下してしまう。

【0010】

このように、キャッシュヒット時の処理速度を向上させようとしてタスク数を少なくするとキャッシュミス時に処理速度が低下し、逆にキャッシュミス時の処理速度を向上させようとしてタスク数を多くするとキャッシュヒット時に処理速度が低下してしまう。なお、ここでいう「キャッシュヒット時及びキャッシュミス時」は、キャッシュメモリに対するものであり、プロセッサキャッシュに対するものではない。

【0011】

そこで、本発明に係るディスクアレイ制御装置では、ディスクアレイのキャッシュヒット時及びキャッシュミス時の双方での処理速度を向上できる構成としている。以下に詳しく説明する。

【0012】

本発明に係るディスクアレイ制御装置は、ホストコンピュータから出力された I/O 処理要求に基づき、キャッシュメモリを有するディスクアレイに対してマルチタスク処理を実行する手段と、各タスク処理の実行に必要なプログラムや制御情報を使用頻度の高いものに入れ換えつつ保持する記憶部とを備えたものである。そして、本発明に係るディスクアレイ制御装置は、ホスト I/O 受信手段、低優先度 I/O 処理実行手段、高優先度 I/O 処理実行手段、実行タスク選択手段を備えたことを特徴とする。ホスト I/O 受信手段は、ホストコンピュータからの I/O 処理要求を受信する。低優先度 I/O 処理実行手段及び高優先度 I/O 処理実行手段はホスト I/O 処理を実行する。低優先度 I/O 処理実行手段が実行する I/O 処理を「低優先タスク処理」、高優先度 I/O 処理実行手段が実行する I/O 処理を「高優先タスク処理」とそれぞれ呼ぶこととする。すなわち、スイッチ低優先度 I/O 処理実行手段は、低優先タスク処理を実行する。高優先度 I/O 処理実行手段は、低優先タスク処理よりも優先的に高優先タスク処理

を実行する。実行タスク選択手段は、ホスト I/O 受信手段で受信された I/O 処理を、実行中の高優先タスク処理の数が一定数未満の場合は高優先タスク処理とし、実行中の高優先タスク処理の数が一定数に一致する場合は低優先タスク処理とする。

【0013】

実行中の高優先タスク処理の数が一定数未満の場合、ホスト I/O 受信手段で受信された I/O 処理は、実行タスク選択手段によって高優先タスク処理とされる。一方、実行中の高優先タスク処理の数が一定数に一致する場合、ホスト I/O 受信手段で受信された I/O 処理は、実行タスク選択手段によって低優先タスク処理とされる。これらの高優先タスク処理及び低優先タスク処理はそれぞれ高優先度 I/O 処理実行手段及び低優先度 I/O 処理実行手段で実行されるが、高優先タスク処理は低優先タスク処理よりも優先的に実行される。

【0014】

従来技術では、タスクに優先度を設けていないので、キャッシュヒット時でも多数のタスクが順次動作する。このため、タスクの実行に必要な制御情報を保持する記憶部において、データの追い出しが発生する。したがって、次に同じタスクが処理を実行しようとしたときに、記憶部にヒットしないのでアクセスに時間がかかる。

【0015】

これに対し、本発明では、一定数の高優先タスクを優先的に実行する。したがって、少数のタスクのみが優先的に実行されることにより、キャッシュヒット時に記憶部においてデータの追い出しが発生しにくいので、処理速度が向上する。一方、キャッシュミス時は、一つの処理に長時間かかるため、実行中のタスク数が多くなる。なぜなら、高優先タスクの占める一定数を越えたタスクは、低優先タスクとして実行させるからである。そのため、キャッシュミス時に実行するタスク数を制限する場合に比べて、多くのタスクが実行されるので、処理速度が向上する。

【0016】

また、低優先タスクのタスク優先度を処理実行開始直後に動的に高優先タスク

と同じ優先度に変更し、処理実行完了時に元の優先度に戻す、タスク優先度変更手段を更に備えた、としてもよい。換言すると、低優先タスク処理の実行直後に当該低優先タスク処理を高優先タスク処理に変更するタスク優先度変更手段を更に備えた、としてもよい。

【0017】

タスクの優先度は、複数のタスクが実行中の場合のタスクの実行順序に影響する。つまり、優先度の高いタスクと優先度の低いタスクとが同時に処理を実行していた場合、優先度の高いタスクばかりが処理を実行し、優先度の低いタスクがいつまでたっても処理を実行できないという現象が発生することがある。これを回避するために、本発明では、低優先タスクの処理実行中はその優先度を高優先タスクと同じにする。

【0018】

ここで、リアルタイムOS等によるマルチタスク処理について簡単に説明しておく（ただし、本発明をリアルタイムOSに限定する主旨ではない。）。ある処理を実行するタスクは、最初は通常「実行待ち状態」にあり、他のあるタスクからの起動指示を待っている。複数のタスクが実行待ち状態にある場合はタスクの「実行待ち行列」が生成され、タスクの起動指示を受けたときにその待ち行列の先頭に並んでいるタスクが処理を引き受け実行する。本発明での「実行待ち行列」では、待ちに入った順番に関係なくタスクの優先度順に、タスクが待ち行列に並ぶ（優先度が同じ場合は待ちに入った順番となる）。

【0019】

このようなマルチタスク処理において、タスクの優先度は、通常、プログラムの初期化時に固定で設定されるため、動作中に変化することはない。そこで、本発明では、低優先タスクが処理を実行開始直後にその優先度を変更し、処理実行を終了して待ち行列に並ぶ直前に優先度を元に戻す。これが「動的」に優先度を変更するという意味である。

【0020】

また、タスクの優先度は待ち行列の並び順に影響するだけでなく、複数のタスクが実行中の場合のタスクの実行順序にも影響する。つまり、優先度の高いタス

クと優先度の低いタスクとが同時に処理を実行していた場合、優先度の高いタスクばかりが処理を実行し、優先度の低いタスクがいつまでたっても処理を実行できないという現象が発生することがある。これを回避するために、本発明では、低優先タスクの処理実行中はその優先度を高優先タスクと同じにする。

【 0 0 2 1 】

更にまた、実行されたタスク処理がキャッシュメモリに対するものである場合にキャッシュヒットと判定するキャッシュヒット判定手段と、一定時間内に実行されたタスク処理の総数に占めるキャッシュヒットの数をキャッシュヒット率とし、このキャッシュヒット率が高ければ実行するタスク処理数を少なくしキャッシュヒット率が低ければ実行するタスク処理数を多くするキャッシュヒット率監視手段とを更に備えた、としてもよい。

【 0 0 2 2 】

キャッシュヒット率が高い場合にタスク処理数を少なくすることにより、実行されるタスク数を確実に少なくできるので、キャッシュヒット時に記憶部においてデータの追い出しがより発生しにくくなり、これにより処理速度がより向上する。一方、キャッシュヒット率が低い場合はタスク処理数を多くすることにより、実行されるタスク数を確実に多くできるので、処理速度がより向上する。

【 0 0 2 3 】

本発明に係るディスクアレイ制御方法（請求項 4 乃至 6）は、本発明に係るディスクアレイ制御装置に用いられるものであり、それぞれ請求項 1 乃至 3 記載の構成に対応する。本発明に係るディスクアレイ制御プログラム（請求項 7 乃至 9）は、本発明に係るディスクアレイ制御装置に用いられるものであり、それぞれ請求項 1 乃至 3 記載の構成に対応する。

【 0 0 2 4 】

換言すると、本発明は次の（１）～（３）ような特徴点を有する。（１）マルチタスク処理において、高優先タスクと低優先タスクとの２種類を用意することにより、プロセッサキャッシュを効率的に使用する点。（２）タスク優先度を動的に変更することにより、低優先タスクの処理遅延を防止する点。（３）キャッシュヒット率を監視することにより、タスクの起動数を調整してプロセッサキャ

ッシュを効率的に使用する点。すなわち、本発明は、マルチタスク処理を行うディスクアレイにおいて、ホスト I/O 受信手段、実行タスク選択手段、高優先度 I/O 処理実行手段（高優先タスク）、低優先度 I/O 処理実行手段（低優先タスク）、タスク優先度変更手段、キャッシュヒット判定手段、キャッシュヒット率監視手段等を備えることにより、プロセッサキャッシュを効率的に使用し、I/O 処理（書き込み／読み出し処理）の高速化を図ることを特徴とする。

【0025】

【発明の実施の形態】

以下、本発明に係るディスクアレイ制御装置の実施形態について、図面を参照しつつ説明する。ただし、特許請求の範囲における「ディスクアレイ制御装置」及び「記憶部」は、それぞれ「ホスト I/F（インタフェース）制御部」及び「プロセッサキャッシュ」と具体化して言い換えるものとする。なお、本発明に係るディスクアレイ制御方法及びディスクアレイ制御プログラムについては、本発明に係るディスクアレイ制御装置に使用されるものであるので、本発明に係るディスクアレイ制御装置の実施形態を説明する中で同時に説明する。

【0026】

図1は、本発明に係るディスクアレイ制御装置のハードウェア構成の一例を示すブロック図である。図2は、本発明に係るディスクアレイ制御装置の第一実施形態を示す機能ブロック図である。以下、これらの図面に基づき説明する。

【0027】

まず、概略的に説明する。図1に示すように、ディスクアレイ20は、ホスト I/F 制御部21、キャッシュメモリ22、ディスク I/F 制御部23、複数のディスク装置24等から構成されている。ホスト I/F 制御部23は、マイクロプロセッサ31、プロセッサキャッシュ32及びローカルメモリ33を含んで構成され、ホストコンピュータ10から出力された I/O 処理要求に基づきディスクアレイ20に対してマルチタスク処理を実行する。プロセッサキャッシュ32は、各タスク処理の実行に必要なプログラムや制御情報を、使用頻度の高いものに入れ換えつつ保持する。

【0028】

図2に示すように、ホストI/F制御部21は、ホストコンピュータ10からのI/O処理要求を受信するホストI/O受信手段40と、低優先タスク処理を実行する低優先度I/O処理実行手段43と、低優先タスク処理よりも優先的に高優先タスク処理を実行する高優先度I/O処理実行手段42と、実行タスク選択手段41とを備えている。実行タスク選択手段41は、ホストI/O受信手段40で受信されたI/O処理を、実行中の高優先タスク処理の数が一定数未満の場合は高優先タスク処理とし、実行中の高優先タスク処理の数が一定数に一致する場合は低優先タスク処理とする。また、本実施形態のホストI/F制御部21は、低優先タスク処理の実行直後に当該低優先タスク処理を高優先タスク処理に変更するタスク優先度変更手段44を更に備えている。

【0029】

更に詳しく説明する。ディスクアレイ20は、ホストコンピュータ10からのI/O（書き込み／読み出し）要求をホストI/F制御部21で処理する。ホストI/F制御部21は、マルチタスク処理によりI/O処理を実行する。マルチタスク処理とは、複数のタスク（ある一連の処理を実行するモジュール）がそれぞれ独立に動作し、同時に多数のタスクが動作することにより、処理能力を向上させるものである。一般にマルチタスク処理においては、それぞれのタスクが個別にメモリ（タスクメモリ）を必要とするため、プロセッサキャッシュ32を搭載して処理の高速化を図っている。この場合、多数のタスクが動作すると、プロセッサキャッシュ32内でのデータの追い出しや再読み込みが発生することにより、最適な性能（処理速度）が得られない場合がある。

【0030】

ディスクアレイ20がホストコンピュータ10から書き込み要求を受けると、ホストI/F制御部21は、キャッシュメモリ22にデータを書き込んだ後、ホストコンピュータ10に対して完了報告を行う。また、ディスクI/F制御部23は、ホストコンピュータ10からの要求と非同期に、キャッシュメモリ22に格納された書き込みデータをディスク装置24に書き込む。また、ディスクアレイ20がホストコンピュータ10から読み出し要求を受けると、ホストI/F制御部21は、キャッシュメモリ22内に有効なデータが存在するか否かを調査し

、存在する場合（キャッシュヒット）にキャッシュメモリ 22 内のデータをホストコンピュータ 10 に送信する。一方、読み出し要求時にキャッシュメモリ 22 内に有効なデータが存在しない場合（キャッシュミス）、ホスト I/F 制御部 21 は、ディスク I/F 制御部 23 に対してディスク装置 24 からのデータ読み出し指示を通知する。通知を受けたディスク I/F 制御部 23 は、要求されたデータをディスク装置 24 からデータを読み出してキャッシュメモリ 22 内に書き込み、ホスト I/F 制御部 21 に完了報告を行う。ディスク I/F 制御部 23 から完了報告を受けたホスト I/F 制御部 21 は、キャッシュメモリ 22 に格納されたデータをホストコンピュータ 10 に送信して完了報告を行う。

【0031】

ホスト I/F 制御部 21 は、ローカルメモリ 33 を有しており、マイクロプロセッサ 31 を動作させるためのマイクロコードと I/O 処理を実行するための制御情報とを保持している。また、マイクロプロセッサ 31 には、処理の高速化を図るためにプロセッサキャッシュ 32 が接続されている。つまり、ローカルメモリ 33 内の同じメモリ領域を何度もアクセスする場合はアクセス速度の速いプロセッサキャッシュ 32 上にデータを移し、2 回目のアクセス以降ではプロセッサキャッシュ 32 上のデータを使用することにより、ローカルメモリ 33 へのアクセス回数を減らして性能を向上させている。プロセッサキャッシュ 32 へのデータ格納方式としては、LRU（Least Recently Used）方式が使用されている。LRU 方式とは、プロセッサキャッシュ 32 にデータを格納するブロックがない場合に、プロセッサキャッシュ 32 上で最後に参照されてから現在までの経過時間がもっとも長いデータブロックを追い出す方式である。

【0032】

次に、言葉を換えてもう一度説明する。

【0033】

ホスト I/F 制御部 21 を機能毎にブロック化したものが図 2 である。図 2 に示すように、ホスト I/F 制御部 21 は、ホスト I/O 受信手段 40、実行タスク選択手段 41、高優先度 I/O 処理実行手段（複数の高優先タスク A）42、低優先度 I/O 処理実行手段（複数の低優先タスク B）43、及びタスク優先度

変更手段 44 を備えている。ホスト I/O 受信手段 40 は、ホストコンピュータ 10 からの I/O 要求を受信して解析する。高優先タスク A 及び低優先タスク B はそれぞれ複数存在し、それぞれのタスクが独立に I/O 処理を実行する。ここで、低優先タスク B については、高優先タスク A よりもタスク優先度が 1 つ低くなるように設定しておく。タスクの優先度が異なる場合、優先度の高いタスクの方が優先度の低いタスクよりも優先して処理される。実行タスク選択手段 41 は、ホスト I/O 受信手段 40 が解析した I/O 処理要求を、高優先度 I/O 処理実行手段 42 又は低優先度 I/O 処理実行手段 43 に送信する。タスク優先度変更手段 44 は、低優先タスク B のタスク優先度を動的に変更する。

【0034】

ホスト I/O 受信手段 40 はホストコンピュータ 10 からの I/O 要求を受信して内容を解析し、I/O 処理要求を実行タスク選択手段 41 に送信する。I/O 処理要求を受信した実行タスク選択手段 41 は、タスク優先度の高い順に高優先度 I/O 処理実行手段 42 及び低優先度 I/O 処理実行手段 43 に処理を渡していく。つまり、I/O 処理を実行していない高優先タスク A が存在する場合は高優先度 I/O 処理実行手段 42 に処理を渡し、全ての高優先タスク A が I/O 処理を実行中の場合は低優先度 I/O 処理実行手段 43 に処理を渡す。ここで、低優先タスク 42 はタスクの優先度を高優先タスク 43 よりも 1 つ低く設定してある。高優先タスク A 及び低優先タスク B はそれぞれ独立にデータ転送等の I/O 処理を実行する。

【0035】

これにより、I/O 処理が短い時間で実行できるような場合（キャッシュヒット時）は、高優先タスク A のみで処理可能であるので、I/O 処理を実行するタスク数が限られる。そのため、常にプロセッサキャッシュ 32 にヒットする状態で処理を実行できるので、処理速度が向上する。また、I/O 処理に長い時間を要するような場合（キャッシュミス時）は、低優先タスク B も含めた多数のタスクが動作する。そのため、プロセッサキャッシュ 32 にヒットしない状態となるものの、多数のタスクが多重に動作することにより処理速度が向上する。

【0036】

ただし、高優先タスクAと低優先タスクBとが同時に処理を実行している場合、タスクの優先度が低いために低優先タスクBの処理が遅延することにより、低優先タスクBが実行しているI/O処理に対するホストコンピュータ10への応答が遅くなるという問題が発生する。場合によってはI/Oエラーに至る可能性がある。これを回避するために、低優先度I/O処理実行手段43は、タスク優先度変更手段44によって自タスクの優先度を動的に変更させる。すなわち、タスク優先度変更手段44は、低優先タスクBの動作開始時にタスクの優先度を1つ上げて高優先タスクAと同じタスク優先度に変更し、低優先タスクBがI/O処理を完了した時にタスクの優先度を1つ下げて元の優先度に戻しておく。これにより、低優先タスクBのI/O処理が遅延することを回避することができる。

【0037】

図3は、図2のホストI/F制御部におけるI/O処理実行タスクの生成処理を示すフローチャートである。図4は、図2のホストI/F制御部におけるタスク優先度変更手段の動作を示すフローチャートである。以下、図1乃至図4に基づき、ホストI/F制御部21の動作を説明する。

【0038】

最初に、I/O処理実行タスクの生成処理について、図4のフローチャートを参照して説明する。まず、タスク番号を0とする(S12)。続いて、タスク優先度をNとしてタスクを生成し(S13)、タスク番号をインクリメントする(S14)。これを高優先タスクAの一定数分繰り返して高優先タスクAを生成する(S15)。続いて、タスク優先度をN-1としてタスクを生成し(S16)、タスク番号をインクリメントする(S17)。これをタスク番号が全タスク数と一致するまで繰り返して、低優先タスクBを生成する(S18)。これにより、優先度がNの高優先タスクAと優先度がN-1の低優先タスクBとが、それぞれ複数生成される。

【0039】

次に、ホストコンピュータ10からのI/O処理要求を受信した時の動作について、図1及び図2を参照して説明する。ホストコンピュータ10からのI/O処理要求を受信したホストI/O受信手段40は、I/O処理内容を解析して実

行タスク選択手段 41 に送信する。I/O 処理要求を受信した実行タスク選択手段 41 は、タスク優先度の高い順に高優先度 I/O 処理実行手段 42 及び低優先度 I/O 処理実行手段 43 に処理を渡していく。つまり、I/O 処理を実行していない高優先タスク A が存在する場合は高優先度 I/O 処理実行手段 42 にタスク処理を渡し、一定数全ての高優先タスク A が I/O 処理を実行中の場合は低優先度 I/O 処理実行手段 43 にタスク処理を渡す。これにより、I/O 処理が短い時間で実行できるような場合（キャッシュヒット時）は、高優先タスク A のみで処理可能である。そのため、I/O 処理を実行するタスク数が限られるため、常にプロセッサキャッシュ 32 にヒットする状態で処理を実行できるので、処理速度が向上する。また、I/O 処理に長い時間を要するような場合（キャッシュミス時）は、低優先タスク B も含めた多数のタスクが動作する。そのため、プロセッサキャッシュ 32 にヒットしない状態となるものの、多数のタスクが多重に動作することにより処理速度が向上する。

【0040】

ただし、高優先タスク A と低優先タスク B とが同時に処理を実行している場合、タスクの優先度が低いために低優先タスク B の処理が遅延し、低優先タスク B が実行している I/O 処理に対するホストコンピュータ 10 への応答が遅くなるという問題が発生する。場合によっては I/O エラーに至る可能性がある。これを回避するため、タスク優先度変更手段 44 により低優先タスク B のタスク優先度を動的に変更する。そのフローチャートを図 3 に示す。

【0041】

図 3 において、実行タスク選択手段 41 から I/O 処理要求を受信した高優先度 I/O 処理実行手段 42 及び低優先度 I/O 処理実行手段 43 は（S01）、自タスクのタスク番号を調査して低優先タスク B か高優先タスク A かを判定する（S02）。自タスクが低優先タスク B の場合は、自タスクの優先度を N に変更する（S03）。続いて、データ転送等の I/O 処理を実行し（S04）、自タスクが低優先タスク B の場合はタスクの優先度を N-1 に戻す（S05, S06）。最後に、I/O 処理を終了して再度 I/O 処理要求待ち状態に入る（S07）。これにより、タスクの優先度が低いことにより、低優先タスク B が実行する

I/O処理が遅延することは、回避される。このようにして、ディスクアレイ 20 のキャッシュヒット性能及びキャッシュミス性能の双方を向上できる。

【0042】

図5は、本発明に係るディスクアレイ制御装置の第二実施形態を示す機能ブロック図である。以下、図1及び図5に基づき説明する。ただし、図5において図2と同じ部分は、同じ符号を付すことにより説明を省略する。

【0043】

本実施形態のホスト I/F 制御部 30 は、ホスト I/O 受信手段 40、実行タスク選択手段 41、複数の高優先度 I/O 処理実行手段（高優先タスク A）42、複数の低優先度 I/O 処理実行手段（低優先タスク B）43、タスク優先度変更手段 44、キャッシュヒット判定手段 45、キャッシュヒット率監視手段 46 等を備えている。

【0044】

ホスト I/O 受信手段 40 は、ホストコンピュータ 10 からの I/O 要求を受信し解析する。高優先タスク A 及び低優先タスク B はそれぞれ複数存在し、それぞれのタスクが独立に I/O 処理を実行する。ここで、低優先タスク B については、高優先タスク A よりもタスク優先度を 1 つ低く設定しておく。タスクの優先度が異なる場合、優先度の高いタスクの方が優先度の低いタスクよりも優先して処理される。実行タスク選択手段 41 は、ホスト I/O 受信手段 40 から受信した I/O 処理要求を高優先度 I/O 処理実行手段 42 又は低優先度 I/O 処理実行手段 43 に送信する。タスク優先度変更手段 44 は、低優先タスク B のタスク優先度を動的に変更する。キャッシュヒット判定手段 45 は、ホスト I/O がキャッシュヒットしているか否かを判定する。キャッシュヒット率監視手段 46 は、ある一定時間内のキャッシュヒット率（キャッシュヒットした I/O 数 ÷ 全体の I/O 処理数）を監視する。

【0045】

次に、ホスト I/F 制御部 30 の動作について説明する。ホストコンピュータ 10 からの I/O 処理要求を受信したホスト I/O 受信手段 40 は、I/O 処理内容を解析して実行タスク選択手段 41 に送信する。一方、キャッシュヒット判

定手段 45 は、ホスト I/O 受信手段 40 で受信したホスト I/O がキャッシュヒットしているか否かを判定する。キャッシュヒットとは、ホスト I/O に該当するアドレスのデータがキャッシュメモリ 22 上に存在することを示す。キャッシュヒットしている場合には、ディスク装置 24 へのアクセスを行う必要がないため、短時間で I/O 処理を実行することができる。キャッシュヒット率監視手段 46 は、キャッシュヒット判定手段 45 の判定結果を使用してある一定時間内のキャッシュヒット率（キャッシュヒットした I/O 数 ÷ 全体の I/O 処理数）を監視し、そのキャッシュヒット率を実行タスク選択手段 41 に通知する。

【0046】

ホスト I/O 受信手段 40 から I/O 処理要求を受信した実行タスク選択手段 41 は、キャッシュヒット率監視手段 43 から通知されたキャッシュヒット率の値に応じて、高優先度 I/O 処理実行手段 42 及び低優先度 I/O 処理実行手段 43 の起動数を調整する。つまり、通知されたキャッシュヒット率がある規定値以上の場合（I/O 処理を短時間で処理することができる場合）は高優先タスク A のみで動作するように、高優先タスク A の一定数を越えて I/O 処理を起動しないように制限する。キャッシュヒット率がある規定値未満の場合（I/O 処理に時間を要する場合）は、低優先タスク B も動作するように全ての I/O 処理を起動する。

【0047】

これにより、キャッシュヒット時は、高優先タスク A のみで処理可能であり、I/O 処理を実行するタスク数が限られる。そのため、常にプロセッサキャッシュ 32 にヒットする状態で処理を実行できるので、処理速度が向上する。また、キャッシュミス時は低優先タスク B も含めた多数のタスクが動作するのでプロセッサキャッシュ 32 にヒットしない状態となるものの、多数のタスクが多重に動作することにより処理速度が向上する。このようにして、ディスクアレイ 20 のキャッシュヒット性能及びキャッシュミス性能の双方を向上できる。

【0048】

【発明の効果】

本発明に係るディスクアレイ制御装置等によれば、実行中の高優先タスク処理

の数が一定数未満の場合はホストコンピュータから受信した I/O 処理を高優先タスク処理とし、実行中の高優先タスク処理の数が一定数に一致する場合はホストコンピュータから受信した I/O 処理を低優先タスク処理とすることにより、少数のタスクのみが優先的に実行されるので、キャッシュヒット時に記憶部においてデータの追い出しが発生しにくくなり、これにより処理速度を向上できる。一方、キャッシュミス時は低優先タスクも含めて実行中のタスク数が多くなることにより、キャッシュミス時に実行するタスク数を制限する場合に比べて、処理速度を向上できる。

【0049】

また、低優先タスク処理を実行直後に高優先タスク処理に変更することにより、低優先タスク処理の遅延を回避できるので、全体としての処理速度を向上できる。

【0050】

更に、キャッシュヒット率が高い場合にタスク処理数を少なくすることにより、実行されるタスク数を確実に少なくできるので、キャッシュヒット時に記憶部においてデータの追い出しがより発生しにくくなり、これにより処理速度をより向上できる。一方、キャッシュヒット率が低い場合はタスク処理数を多くすることにより、実行されるタスク数を確実に多くできるので、処理速度をより向上できる。

【0051】

換言すると、本発明によれば次の効果を奏する。

【0052】

第1の効果は、I/O 処理が短い時間で実行できるような場合（キャッシュヒット時）のディスクアレイの性能を向上できることにある。その理由は、I/O 処理が短い時間で実行できるような場合は、高優先タスクのみで処理可能であるため、常にプロセッサキャッシュにヒットする状態で処理を実行できるからである。

【0053】

第2の効果は、I/O 処理に長い時間を要するような場合（キャッシュミス時

) のディスクアレイの性能を向上できることにある。その理由は、I/O処理に長い時間を要するような場合は、低優先タスクも含めた多数のタスクが動作するので、十分な数のタスクを用意しておけば多数のタスクが多重に動作できるからである。

【0054】

第3の効果は、キャッシュヒット性能及びキャッシュミス性能の双方を向上した上で、ホストコンピュータに対する完了報告を遅延させることなく、各I/O処理の処理時間を均等にできることにある。その理由は、タスクの優先度を動的に変更することにより、I/O処理実行中に全タスクの優先度が同一になるように制御しているからである。

【図面の簡単な説明】

【図1】

本発明に係るディスクアレイ制御装置のハードウェア構成の一例を示すブロック図である。

【図2】

本発明に係るディスクアレイ制御装置の第一実施形態を示す機能ブロック図である。

【図3】

図2のホストI/F制御部におけるI/O処理実行タスクの生成処理を示すフローチャートである。

【図4】

図2のホストI/F制御部におけるタスク優先度変更手段の動作を示すフローチャートである。

【図5】

本発明に係るディスクアレイ制御装置の第二実施形態を示す機能ブロック図である。

【符号の説明】

10 ホストコンピュータ

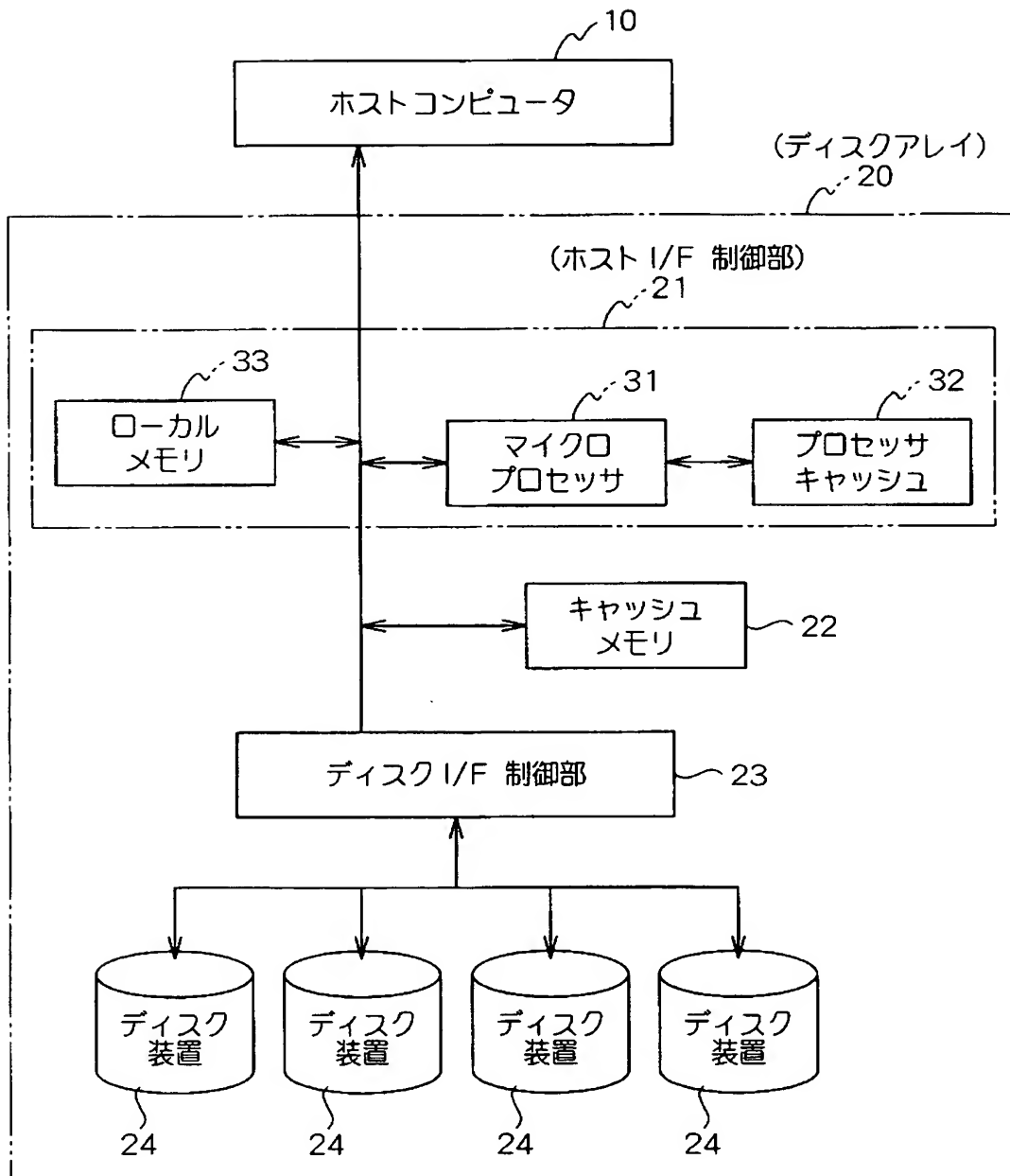
20 ディスクアレイ

- 2 1, 3 0 ホスト I / F 制御部 (ディスクアレイ制御装置)
- 2 2 キャッシュメモリ
- 2 3 ディスク I / F 制御部
- 2 4 ディスク装置
- 3 1 マイクロプロセッサ
- 3 2 プロセッサキャッシュ (記憶部)
- 3 3 ローカルメモリ
- 4 0 ホスト I / O 受信手段
- 4 1 実行タスク選択手段
- 4 2 高優先度 I / O 処理実行手段
- 4 3 低優先度 I / O 処理実行手段
- 4 4 タスク優先度変更手段
- 4 5 キャッシュヒット判定手段
- 4 6 キャッシュヒット率監視手段

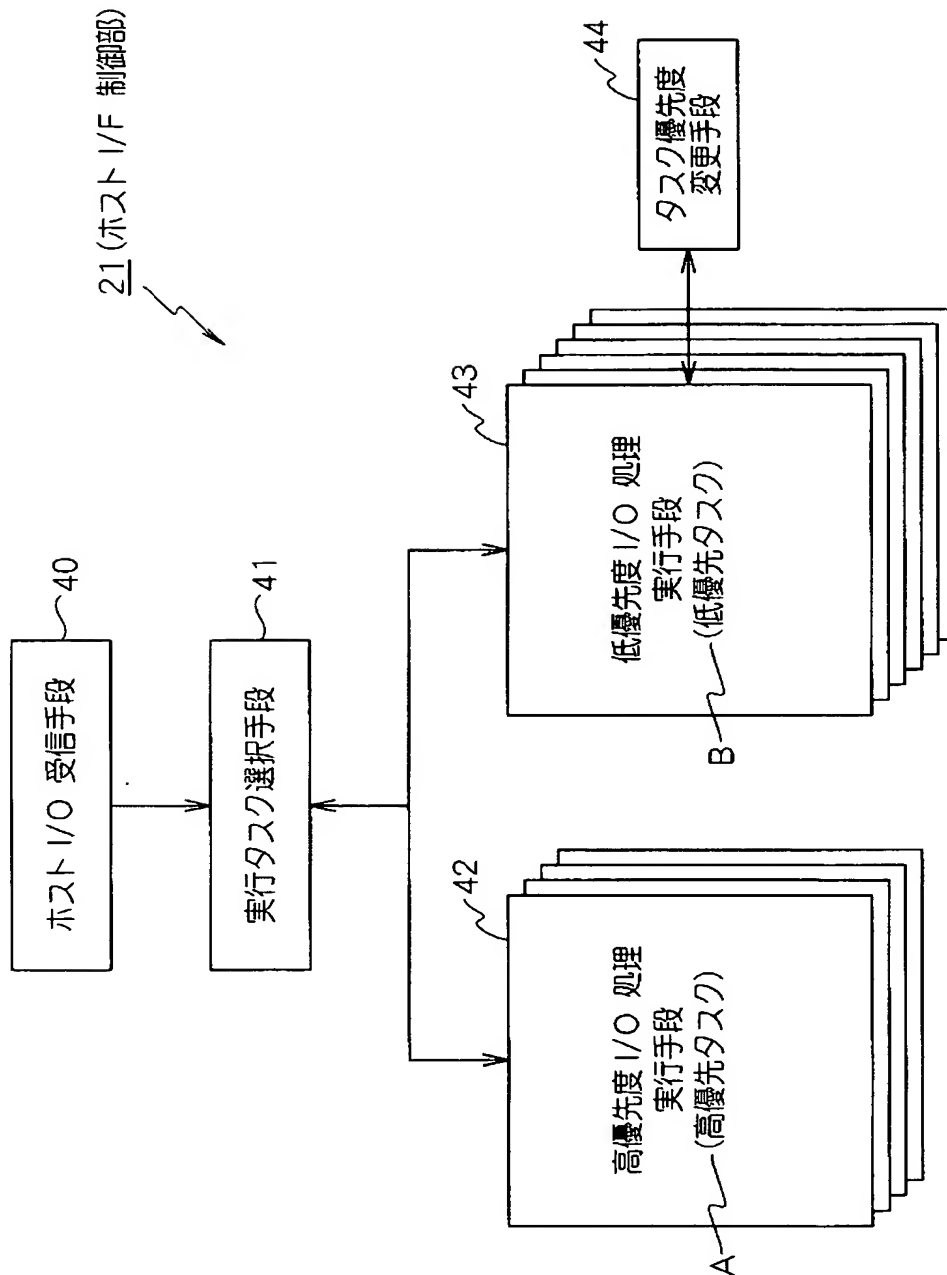
【書類名】

図面

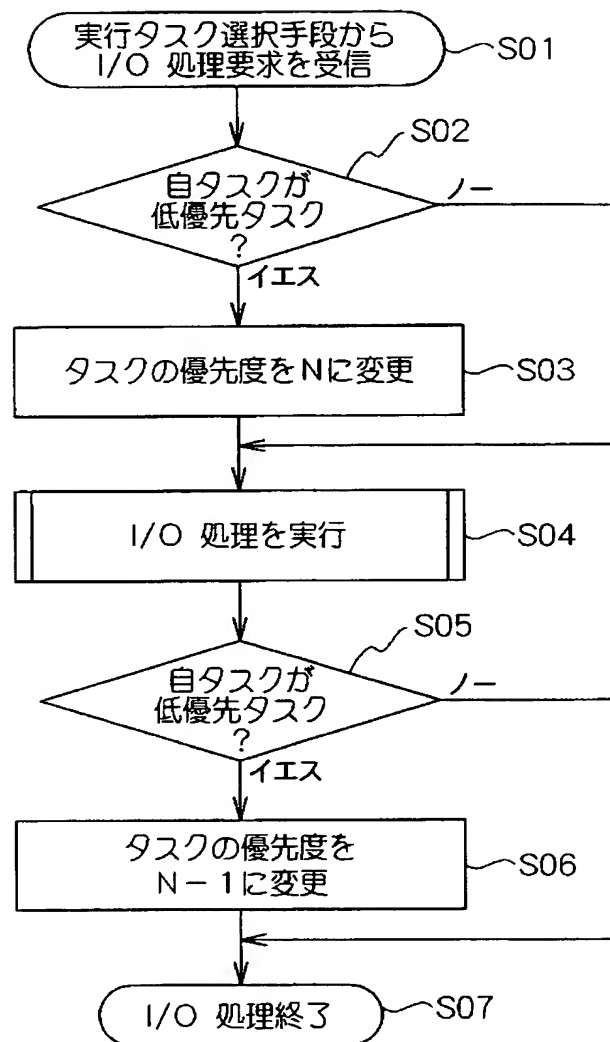
【図 1】



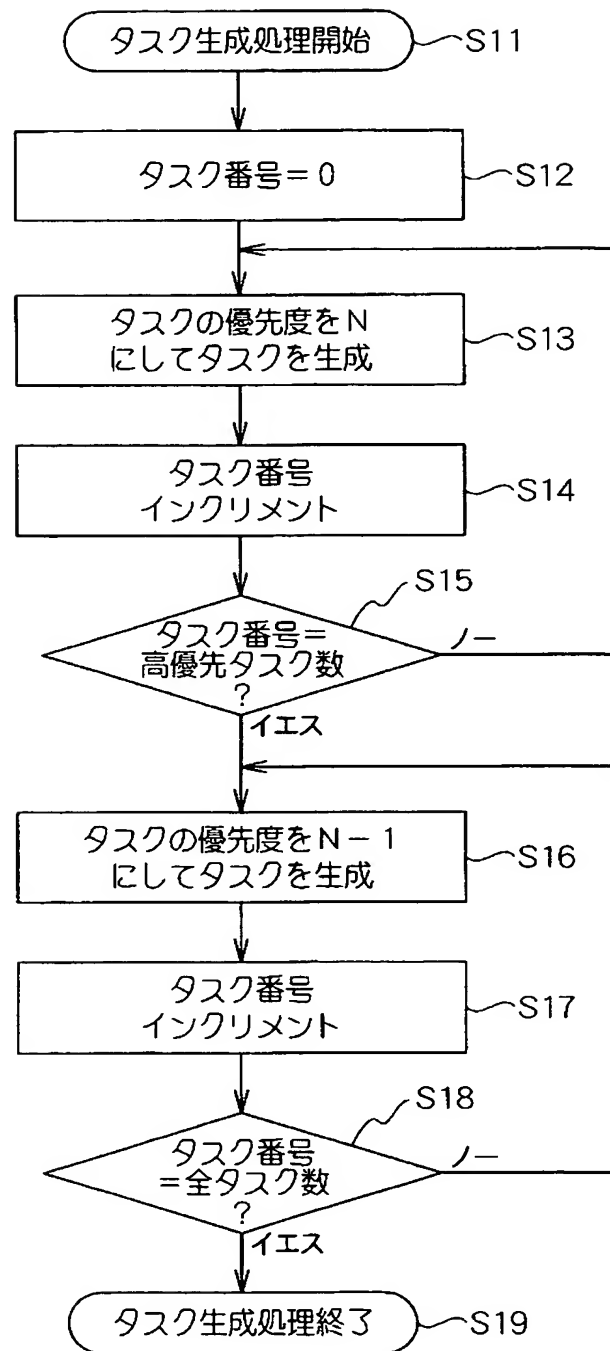
【図 2】



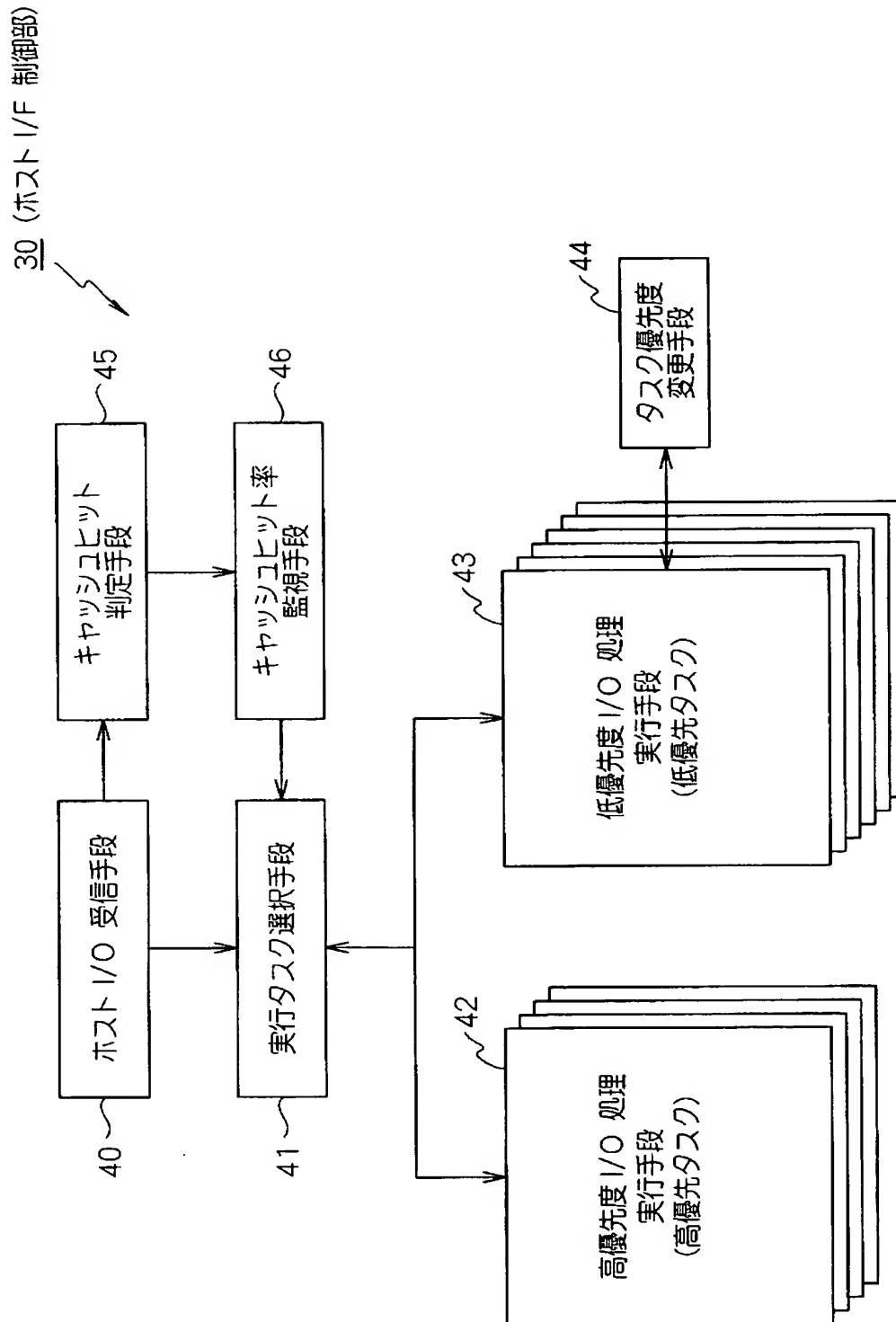
【図 3】



【図 4】



【図 5】



【書類名】 要約書

【要約】

【課題】 ディスクアレイの処理速度を向上させる。

【解決手段】 ホスト I/O 制御部 21 は、ホストコンピュータからの I/O 処理要求を受信するホスト I/O 受信手段 40 と、低優先タスク処理を実行する低優先度 I/O 処理実行手段 43 と、低優先タスク処理よりも優先的に高優先タスク処理を実行する高優先度 I/O 処理実行手段 42 と、実行タスク選択手段 41 とを備えている。実行タスク選択手段 41 は、ホスト I/O 受信手段 40 で受信された I/O 処理を、実行中の高優先タスク処理の数が一定数未満の場合は高優先タスク処理とし、実行中の高優先タスク処理の数が一定数に一致する場合は低優先タスク処理とする。少数のタスクのみが優先的に実行されるので、キャッシュヒット時にプロセッサキャッシュにおいてデータの追い出しが発生しにくくなり、これにより処理速度を向上できる。

【選択図】 図 2

特願 2 0 0 3 - 0 3 9 1 8 5

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 4 2 3 7]

1 . 変更年月日

1 9 9 0 年 8 月 2 9 日

[変更理由]

新規登録

住 所

東京都港区芝五丁目 7 番 1 号

氏 名

日本電気株式会社